Creation of digital terrain models using surface evolution

M. Kollár[1], Z. Minarechová[1] ,K.Mikula[1]
[1]Slovak University of Technology in Bratislava, Dept. of Mathematics- Faculty of Civil Engineering, Bratislava, Slovak Republic
Contact: kollarm@math.sk

Abstract

In our contribution we discuss the creation of the digital terrain models using a surface evolution by the weighted mean curvature flow. To achieve this goal we have developed a discretization of the Laplace Beltrami operator using the finite volume method. This approach is based on an approximation of an arbitrary surface by triangular mesh and deriving the weak formulation for the Laplace-Beltrami operator on the manifold. A system of linear equations obtained by the finite volume approximation of the weak formulation is solved in each discrete time step by an iterative solver. The numerical experiments consist of theoretical ones, where we have tested the proposed approach by finding a minimal surface and a surface with a given mean curvature, and practical ones, where our aim was to create the digital terrain models obtained by using a remote sensing technology LiDAR.

DTM (Digital terrain model)

Digital terrain model is used in engineering applications, when is necessary to create models which will exclude only some technical element of the country or vegetation. There are many ways to create such a model. In our work we presented an option to create DTM by surface evolution. Input data to our model represents data obtained by LiDAR technology.

- **LiDAR** (*Light Detection And Ranging*) stands for optical technology that uses pulsed laser light to measure the distance between the object and the LiDAR device. This technology allows mapping the surface in high resolution.
- **LiDAR** data – flying altitudes + spectral information from aerial images

Surface evolution by mean curvature

Process of surface evolution by mean curvature can be express by parabolic PDE:

$\partial_t \boldsymbol{S}(t) = \Delta_s \boldsymbol{S}(t), \quad t \in [0,T]$ Δ_s - the Laplace-Beltrami operator

- Initial condition : $\boldsymbol{S}(0) = \Omega^0, \quad \Omega^0 \subset \mathbb{R}^3$
- Solution: $\boldsymbol{S}(t) = x(t)\vec{\imath} + y(t)\vec{\jmath} + z(t)\vec{k} = \Omega^t$

Ω - surface (computational domain)

Extension of model by curvature detector:

$\partial_t \boldsymbol{S}(t) = c(v)\Delta_s \boldsymbol{S}(t),$ Main goal is to slow an evolution of surface in points where surface has generally larger mean curvature

- Curvature detector: $c(v) = \dfrac{1}{1+Kv}, \quad K \geq 0, \quad v = |\Delta_s \bar{\boldsymbol{S}}(t)|^2$

K – sensibility coefficeient $\bar{\boldsymbol{S}}(t)$ - solution to previous not-extended model

Surface finite volume method

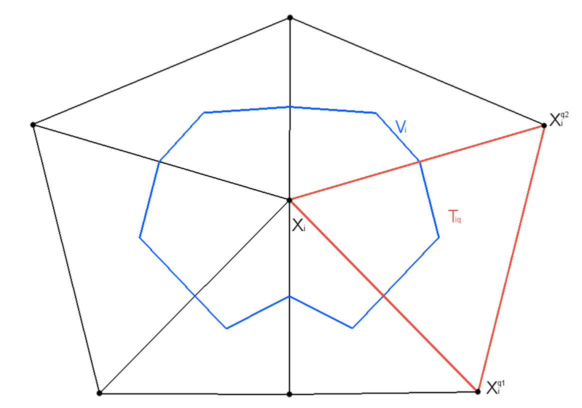
Differential equation was solved by the FVM. Time partial derivation was expressed by Euler backward discretization.

$$\tau_j = t_j - t_{j-1}, \quad \boldsymbol{S}(t_j) = \boldsymbol{S}^j, \quad \partial_t \boldsymbol{S}(t) = \frac{\boldsymbol{S}^j - \boldsymbol{S}^{j-1}}{\tau_j}$$

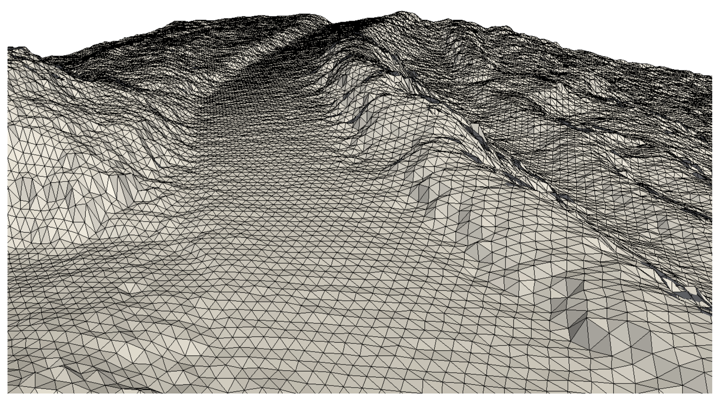
Computational domain was approximated by appropriate triangulation T_{iq}, whose vertices represent node and $\boldsymbol{S}_i^j$ represent solution in this nodes.

$$X_i = \boldsymbol{S}_i^j, \qquad X_i \in \Omega^t, \qquad i = 1, \dots, N,$$

Finite volume V_i is defined as polygon with vertices in triangle mass center and center of connecting lines between central node and it neighbors.



Finite volume V_i



Triangulation sample

Final numerical schemes:

- Semi-implicit numerical scheme for classic model:

$$\boldsymbol{S}_i^j - \frac{\tau_j}{m\left(V_i^{j-1}\right)} \sum_{q=1}^{Q_i} \left[m\left(e_{iq}^{1,j-1}\right) \vec{\eta}_{iq}^{1,j-1} \cdot P_{T_{iq}}^j + m\left(e_{iq}^{2,j-1}\right) \vec{\eta}_{iq}^{2,j-1} \cdot P_{T_{iq}}^j \right] = \boldsymbol{S}_i^{j-1}$$

- Semi-implicit numerical scheme for extended model:

$$\boldsymbol{S}_i^j - \frac{\tau_j}{m\left(V_i^{j-1}\right)} c(|\Delta_s \bar{\boldsymbol{S}}(t)|^2) \sum_{q=1}^{Q_i} \left[m\left(e_{iq}^{1,j-1}\right) \vec{\eta}_{iq}^{1,j-1} \cdot P_{T_{iq}}^j + m\left(e_{iq}^{2,j-1}\right) \vec{\eta}_{iq}^{2,j-1} \cdot P_{T_{iq}}^j \right] = \boldsymbol{S}_i^{j-1}$$

$P_{T_{iq}}^j$ - surface gradient representation on a triangle.

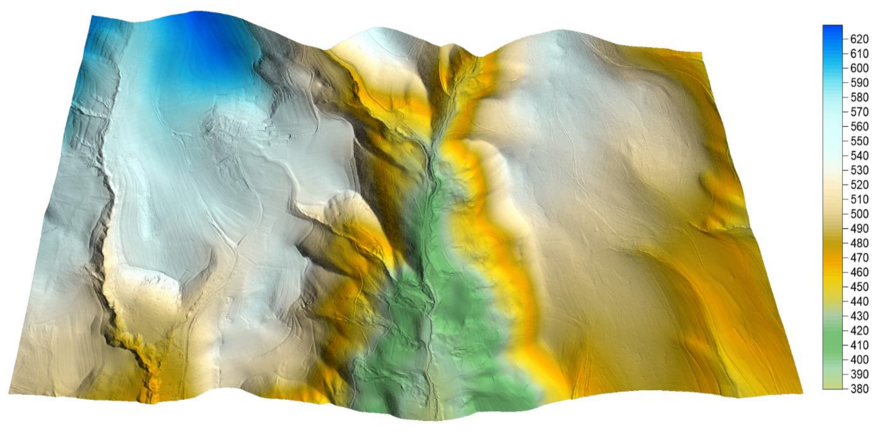
$$P_{T_{iq}}^j = \frac{1}{m(T_{iq}^{j-1})} \left(\frac{\boldsymbol{S}_i^j + \boldsymbol{S}_{q1}^j}{2} d_{iq1}^{j-1} \vec{n}_{iq1}^{j-1} + \frac{\boldsymbol{S}_i^j + \boldsymbol{S}_{q2}^j}{2} d_{iq2}^{j-1} \vec{n}_{iq2}^{j-1} + \frac{\boldsymbol{S}_{q1}^j + \boldsymbol{S}_{q2}^j}{2} d_{q1q2}^{j-1} \vec{n}_{q1q2}^{j-1} \right)$$

Numerical experiment

- Input data: DSM model with partially removed vegetation obtained by LiDAR technology.
- Goal: Remove vegetation fragments, noise but preserve road structures

Discretization	$0.5\text{m} \times 0.5\text{m}$
Nodes	15 008 001
Triangles	30 000 000

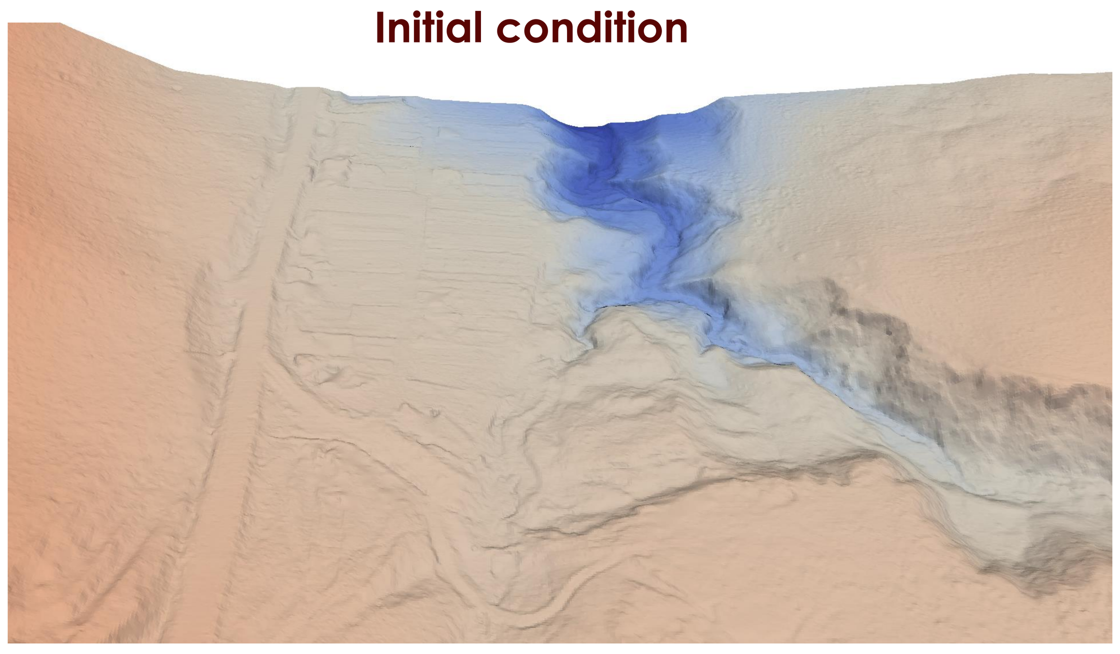
Model statistics

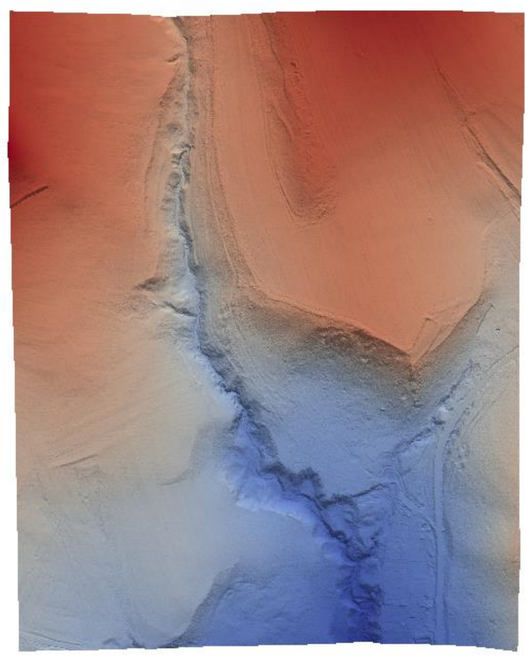


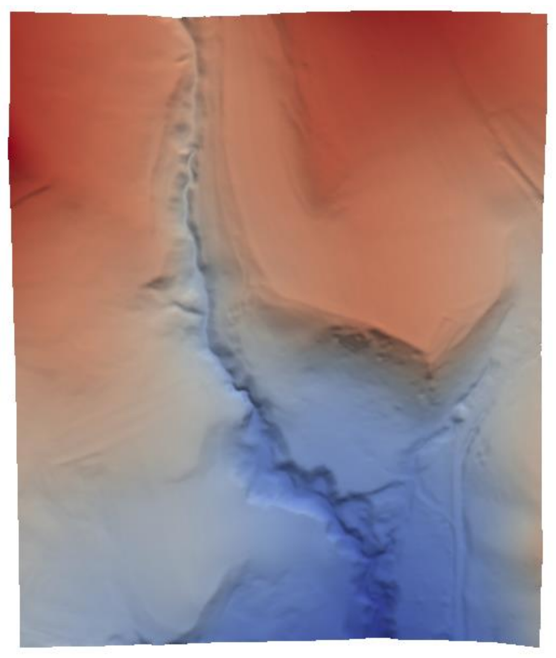
Input data

Results of numerical experiment

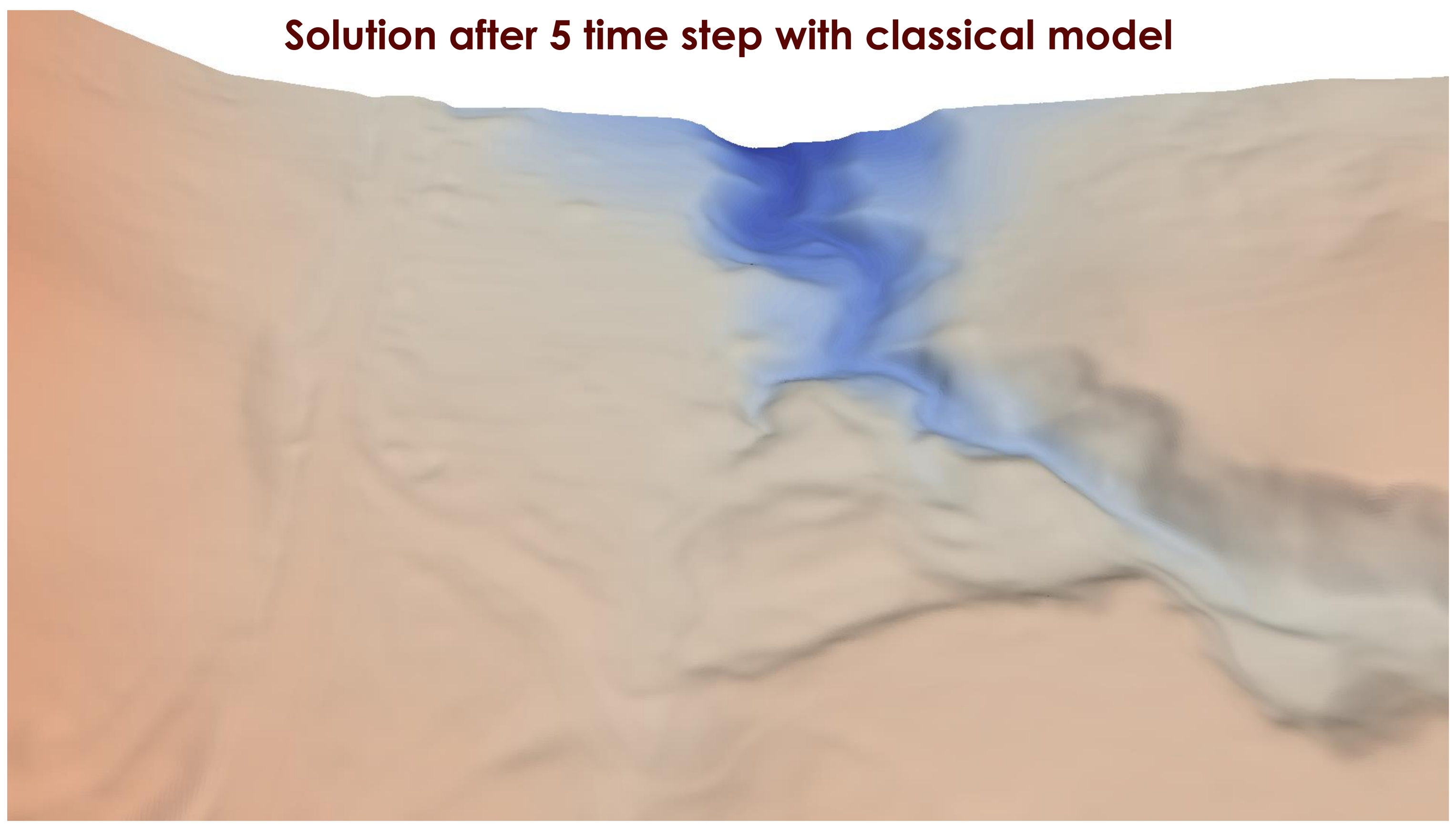
Initial condition







Solution after 5 time step with classical model



Solution after 5 time step with extended model, K=100